

图文 072、大厂面试题：什么是内存溢出？在哪些区域会发生内存溢出？

874 人次阅读 2019-09-10 07:00:00

详情 评论

大厂面试题：  
什么是内存溢出？在哪些区域会发生内存溢出？



狸猫技术窝

狸猫技术窝专栏上新，基于**真实订单系统**的消息中间件（mq）实战，重磅推荐：

进店逛



相关频道



从 0 开  
战高手  
已更新1

未来3个月，我的好朋友原子弹大侠将带你一起，全程实战，360度死磕MQ

(点击下方蓝字进行试听)

[从 0 开始带你成为消息中间件实战高手](#)

重要说明：

**如何提问：**每篇文章都有评论区，大家可以尽情在评论区留言提问，我都会逐一答疑

(ps：评论区还精选了一些小伙伴对**专栏每日思考题**的作答，有的答案真的非常好！大家可以通过看别人的思路，启发一下自己，从而加深理解)

**如何加群：**购买了狸猫技术窝专栏的小伙伴都可以加入**狸猫技术交流群**。

(群里有不少**一二线互联网大厂**的助教，大家可以一起讨论交流各种技术)

具体**加群方式**请参见文末。

(注：以前通过其他专栏加过群的同学就不要重复加了)

## 1、前言

这篇文章，我们来聊一个面试常常被问的问题：JVM里的内存溢出到底是指的什么，哪些区域有可能会发生内存溢出？

要搞明白这个问题，那我们就得从头儿开始来捋一下了，我们这篇文章一步一图，从JVM的核心运行原理出发，然后给大家带出来到底哪些地方可能会发生内存溢出。

## 2、运行一个Java系统就是运行一个JVM进程

首先的话呢，大家得先搞明白一个事情，就是我们平时说启动一个Java系统，其实本质就是启动一个JVM进程。

咱们就用最最基本的情况来给大家演示一下好了，比如说下面的一段代码，是每个Java初学者都会写的一段代码：

```
public class HelloWorld {  
  
    public static void main(String[] args) {  
        String message = "Hello World";  
        System.out.println(message);  
    }  
}
```

那么大家知道，当你在Eclipse或者IntelliJ IDEA中写好这个代码，然后通过IDE来运行这个代码的时候，会发生哪些事情吗？

首先，我们专栏最早的几篇文章就给大家说过，我们写好的代码他都是后缀为“.java”的源代码，这个代码是不能运行的。

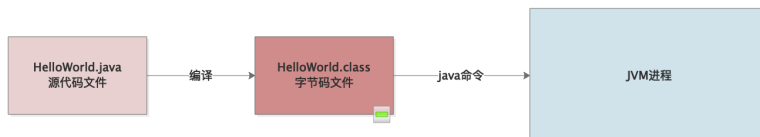
所以第一步就是这份“.java”源代码文件必须先编译成一个“.class”字节码文件，这个字节码文件才是可以运行的，如下图所示。



接着对于这种编译

好的字节码文件，比如HelloWorld.class，如果里面包含了main方法，接下来我们就可以用“java命令”来在命令行执行这个字节码文件了

实际上一旦你执行“java命令”，相当于就会启动一个JVM进程。这个JVM进程就会负责去执行你写好的那些代码，如下图所示。



所以首先要清楚第一点，运行一个Java系统，本质上就是启动一个JVM进程，这个JVM进程负责来执行你写好的一大堆代码。只要你的Java系统中包含一个main方法，接着JVM进程就会从你指定的这个main方法入手，开始执行你写的代码。

## 3、到底执行哪些代码：JVM得加载你写的类

下一个问题，JVM进程怎么执行你写的那些代码呢？

大家都知道，Java是一个面向对象的语言，所以最最基本的代码组成单元就是一个一个的类，平时我们说写Java代码，不就是写一个的一个的类吗？是不是。

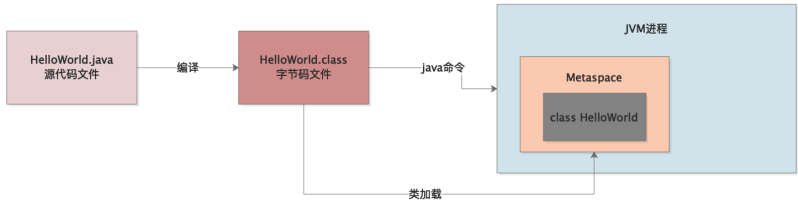
然后在一个个的类里我们会定义各种变量，方法，数据结构，通过if else之类的语法，写出来各种各样的系统业务逻辑，这就是所谓的编程了。

所以JVM既要执行你写的代码，首先当然得把你写好的类加载到内存里来啊！

所以JVM的内存区域里大家都知道，有一块区域叫做永久代，当然JDK 1.8以后都叫做Metaspace了，我们也用最新的说法好了。

这块内存区域就是用来存放你系统里的各种类的信息的，包括JDK自身内置的一些类的信息，都在这块区域里。

JVM有类加载器和一套类加载的机制，我们在专栏最开始的时候都说过了，这里不再赘述，他会负责把我们写好的类从编译好的“.class” 字节码文件里加载到内存里来，如下图。



好，那么既然有这么一块Metaspace区域是用来存放类信息的，**那是不是有可能在这个Metaspace区域里就会发生OOM？**

没错，是有这种可能的。

#### 4、Java虚拟机栈：让线程执行各种方法

大家都知道，我们写好的那些Java代码虽然是一个一个的类，但是其实核心的代码逻辑一般都是封装在类里面的各种方法中的

比如JVM已经加载了我们写好的HelloWorld类到内存里了，接着怎么执行他里面的代码呢？

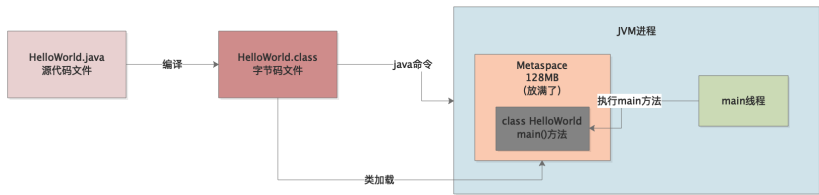
Java语言中的一个通用的规则，就是一个JVM进程总是从main方法开始执行的，所以我们既然在HelloWorld中写了一个main()方法，那么当然得执行这个方法中的代码了。

但是等一等，JVM进程里的谁去执行main()方法的代码？

其实我们所有的方法执行，都必须依赖JVM进程中的某个线程去执行，你可以理解为线程才是执行我们写的代码的核心主体。

JVM进程启动之后默认就会有一个main线程，这个main线程就是专门负责执行main()方法的。

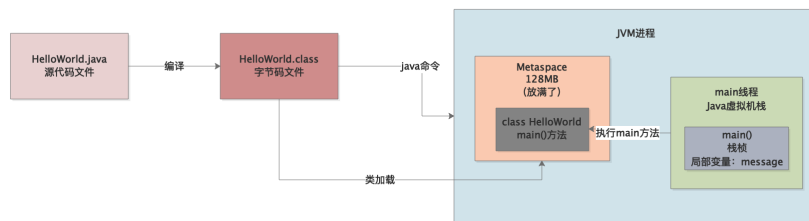
大家如下图所示。



现在又有一个问题了，在main()方法里定义了一个局部变量，“message”，那么大家回忆一下，这些方法里的局部变量可能会有很多，那么这些局部变量是放在哪里的呢？

很简单，每个线程都有一个自己的虚拟机栈，就是所谓的**栈内存**。

然后这个线程只要执行一个方法，就会为方法创建一个栈帧，将栈帧放入自己的虚拟机栈里去，然后在这个栈帧里放入方法中定义的各种局部变量，如下图所示。



好，现在问题来

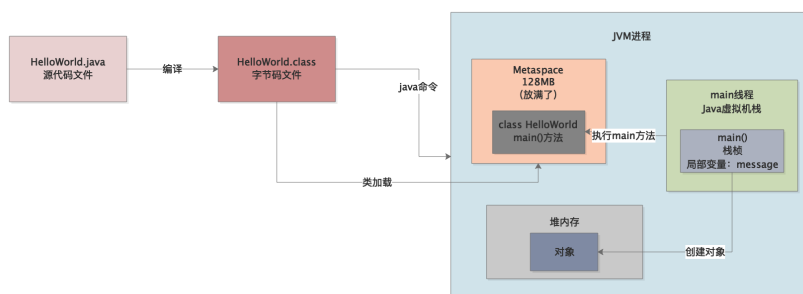
了，大家如果还记得之前我们讲过的一个参数，应该都知道，我们可以设置JVM中每个线程的虚拟机栈的内存大小的，一般是设置为1MB。

那么既然每个线程的虚拟机栈的内存大小是固定的，是否可能会发生虚拟机栈的内存溢出？

没错，所以**第二块可能发生OOM的区域，就是每个线程的虚拟机栈内存。**

## 5、堆内存：放我们创建的各种对象

最后我们知道，我们写好的代码里，特别在一些方法中，可能会频繁的创建各种各样的对象，这些对象都是放在堆内存里的，如下图所示。



而且我们通过之前的学习，也都知道了一点，通常我们在JVM中分配给堆内存的空间其实一般是固定的

既然如此，我们不停在堆内存里创建对象，是不是说明，堆内存也有可能发生内存溢出？

没错，**第三块可能发生内存溢出的区域，就是堆内存空间！**

## 6、本文总结

这篇文章我们从Java代码的运行的角度去分析了一下，我们写好的代码在运行的过程中涉及到了哪几块内存区域，然后这几块内存区域是不是就是有可能发生内存溢出的区域

大家先通过这篇文章的一步一图分析回顾一下，脑子里慢慢形成一些流动的图形，**接下来三天，我们就分别来分析一下Metaspace、Java虚拟机栈和堆内存这几块内存区域到底在什么情况下会发生内存溢出。**

**End**

狸猫技术窝专栏上新，基于**真实订单系统**的消息中间件（mq）实战，重磅推荐：